4Ш

УДК 004.89:004.852:629.4.054.2

В.И. Шинкаренко, Е.Г. Васецкий, И.Д. Корякин
Днепропетровский национальный университет железнодорожного транспорта
имени академика В. Лазаряна, г. Днепропетровск, Украина
ccp@diit-70.dp.ua

Адаптивная фильтрация изображений скоростемерных лент локомотивов

В статье представлены разработанные методы и средства структурной адаптации алгоритмов программной фильтрации изображения. Решается задача выделения и утоньшения линий скоростемерных лент локомотивов. С применением генетических алгоритмов подбирается последовательность базовых фильтров и их параметров, адаптированная к заданным шаблонам линий согласно предложенному критерию.

Введение

Основным средством контроля работы локомотивных бригад является запись на скоростемерной ленте. В настоящее время на локомотивах устанавливаются электронные скоростемеры, однако подавляющее большинство оборудовано механическими [1]. По экономическим причинам период перехода на электронные скоростемеры займет не одно десятилетие.

Расшифровка скоростемерных лент на бумажных носителях выполняется вручную. Для качественной расшифровки скоростемерных лент техники-шифровальщики используют специальные шаблоны, линейки, линзы, планшеты, номограммы участков обслуживания с нанесенными местами расположения станций, светофоров, километровых отметок и т.п. На качество расшифровки в значительной мере влияют профессиональные знания и навыки исполнителей.

Для повышения качества анализа информации разрабатывается программно-аппаратный комплекс с такой функциональностью: перенос информации с бумажного носителя на электронный, анализ растрового изображения, выявление нарушений режима и требований правил управления локомотивом, выполнения необходимых технологических операций управления поездом.

Прагматическая эффективность автоматизации расшифровки скоростемерных лент имеет две составляющие. Во-первых, она дает возможность более полно и качественно выполнять анализ процесса управления поездом. Качественный контроль работы локомотивных бригад повышает дисциплину и сказывается на безопасности движения. Во-вторых, компьютерная обработка отсканированной скоростемерной ленты существенно улучшает качество изображения, что облегчает работу по расшифровке.

Работы по автоматизации ведутся как в Украине [2], так и в России [3], однако ввиду сложности задачи добиться необходимого уровня качества распознавания пока не удается.

В данной работе рассматривается одна из составляющих автоматизированной системы расшифровки скоростемерных лент, связанная с выделением линий на основе программных фильтров [4-8].

Постановка задачи

Автоматизированная обработка скоростемерных лент предполагает решение следующих задач:

– формирования растрового изображения на электронном носителе;

– повышения качества изображения;

– выделения и идентификации линий, нанесенных писцами скоростемера на скоростемерной ленте;

– наложения ленты на шаблон поездки;

– выявления нарушений технологии ведения поезда и регистрируемых неисправностей.

В данной статье решаются задачи повышения качества изображения и выделения линий.

В результате сканирования на разработанном авторами устройстве [2] имеем растровое изображение в виде матрицы:

$$C = [c_{ij}],\ i = 1..I,\ j = 1..J\ ,$$

где c_{ij} – цвет пиксела в точке (i, j) в формате RGB, т.е. $c_{ij} = r_{ij} + g_{ij} + b_{ij}$, а r_{ij}, g_{ij}, b_{ij} – интенсивность красной, зеленой и голубой составляющих соответственно, в пределах 0…255.

Задача упрощается, если изображение переведено в формат Grayscale (оттенки серого):

$$c_{ij}^{*} = [a_1 \cdot R_{ij} + a_2 \cdot G_{ij} + a_3 \cdot B_{ij}]\ ,$$

где c_{ij}^{*} – оттенок серого в точке (i, j) в пределах 0…255, a_i – коэффициенты, а [ ] – целая часть числа (только здесь, в других местах – это обозначение матрицы).

Обозначим x_{ij} – пиксел с координатами *(i, j)* в изображении (матрице) *C*, и массив пикселов $X = [x_{ij}]$, а также, соответственно, x_{ij}^{*} и $X^{*} = [x_{ij}^{*}]$ – в матрице C^{*}.

Изображение X^{*} состоит из точек фона $X_{Ф}$, линий $X_{Л}$, сетки $X_{С}$ и шумов $X_{Ш}$:

$$X^{*} = X_{Ф} \cup X_{Л} \cup X_{С} \cup X_{Ш},\ X_{Ф} \cap X_{Л} = \varnothing\ .$$

При сканировании с разрешением 300 dpi толщина линии варьируется в пределах от 5 до 15 пикселей, что затрудняет определение направления линии при выявлении нарушений. Утоньшение линий до одного пиксела устраняет неопределенность, связанную с толщиной линий. Определим линию толщиной в один пиксел таким образом.

Пусть точки $x_{ij}^{*} \in X_{Ф}$, если $c_{ij}^{*} > \tau$. Обозначим точки фона $\overline{x}_{ij}^{*}$, а их цвет в формате Grayscale – $\overline{c}_{ij}^{*}$. Аналогично точки $\underline{x}_{ij}^{*} \in X_{Л}$, если $\underline{c}_{ij}^{*} < \tau$ для линий, где τ – некоторый порог.

Точка принадлежит некоторой линии, если она лежит в заданных для линии пределах по вертикали $(\underline{j}_K, \overline{j}_K)$ и её цвет ближе к черному:

$$\underline{x}_{ij}^{*} \in L_n : \underline{j}_K \le j \le \overline{j}_K \ \&\ c_{ij} \le \tau\ . \quad (1)$$

Считая расстояние между точками как:

$$\rho(x_{ij}^{*}, x_{kl}^{*}) = \sqrt{(i-k)^2 + (j-l)^2}\ , \quad (2)$$

расстояние от точки линии до фона будет

$$\rho(\underline{x}_{ij}^{*}, X_{Ф}) = \min_{\substack{k \in 1..I \\ l \in 1..J}} \rho(\underline{x}_{ij}^{*}, \overline{x}_{kl}^{*})\ . \quad (3)$$

Пусть список точек фона, ближайших к точке линии $\underline{x}_{ij}^*$:

$$\hat{X}_{\Phi}^*(\underline{x}_{ij}^*) : \rho(\underline{x}_{ij}^*, \overline{x}_{kl}^*) = \min_{\substack{k\in 1..I \\ l\in 1..J}} \rho(\underline{x}_{ij}^*, \overline{x}_{kl}^*) \tag{4}$$

и $\hat{X}_{\Phi}^*(\underline{x}_{ij}^*, n)$ – n-я точка в этом списке, а функции φ_1 и φ_2 определяют индексы этих точек в массиве $\hat{X}_{\Phi}^*(\underline{x}_{ij}^*, n)$

$$\begin{aligned} k = \varphi_1(\underline{x}_{ij}^*, n) : \overline{x}_{kl}^* = \hat{X}_{\Phi}^*(\underline{x}_{ij}^*, n) \\ l = \varphi_2(\underline{x}_{ij}^*, n) : \overline{x}_{kl}^* = \hat{X}_{\Phi}^*(\underline{x}_{ij}^*, n), \end{aligned} \tag{5}$$

тогда точки линии толщиной не более, чем в два пиксела, определяются так:

$$\underline{x}_{ij}^* \in L_n^{(2)} : \underline{j}_K \le j \le \overline{j}_K \;\&\; c_{ij} \le \tau \;\&\; \rho(\underline{x}_{ij}^*, X_{\Phi}) = \max_{\Psi} \rho(\underline{x}_{ij}^*, \overline{x}_{kl}^*), \tag{6}$$

где $\psi \equiv k = | i + a(i - \varphi_1(\underline{x}_{ij}^*, n)) \;\&\; l = | j + a(j - \varphi_2(\underline{x}_{ij}^*, n))$ при $a = \pm 1, \pm 2...$, $n = 1, 2...$ и $k = 1..I$, $l = 1..J$.

Условие (6) аналогично определению точек центров окружностей, вписанных в «толстую» линию, в случае непрерывных границ линий. В (6) определяется середина линии толщиной в два пиксела $L_n^{(2)}$ при четном количестве пикселов по ширине линии L_n. При нечетном количестве пикселов в ширине линии L_n линия $L_n^{(2)}$ будет толщиной в один пиксел, а так как толщина реальных линий постоянно изменяется, то в общем случае можно говорить, что линия $L_n^{(2)}$ толщиной не более двух пикселей.

Обозначим $\hat{X}_{Л}^*(\underline{x}_{ij}^*)$ – множество точек линии, соседних с $\underline{x}_{ij}^*$ и $\nu(X)$ – количество элементов множества X, тогда

$$\underline{x}_{kl}^* \in \hat{X}_{Л}^*(\underline{x}_{ij}^*) : \underline{x}_{kl}^* \in L_n^{(2)} \;\&\; \rho(\underline{x}_{kl}^*, \underline{x}_{ij}^*) \le \sqrt{2} \;\&\; (k \ne i \,|\, l \ne j) \tag{7}$$

и непрерывную линию толщиной в один пиксел можно определить таким образом:

$$\begin{aligned} &\underline{x}_{ij}^* \in L_n^{(1)} : \underline{x}_{ij}^* \in L_n^{(2)} \;\&\; 1 \le \nu(\hat{X}_{Л}^*(\underline{x}_{ij}^*)) \le 4 \;\& \\ &(\forall \underline{x}_{ij}^* : \nu(\hat{X}_{Л}^*(\underline{x}_{ij}^*)) = 3 \Rightarrow \forall \underline{x}_{kl}^* \in \hat{X}_{Л}^*(\underline{x}_{ij}^*) \Rightarrow \nu(\hat{X}_{Л}^*(\underline{x}_{kl}^*)) < 3). \end{aligned} \tag{8}$$

Согласно (8) у каждой точки линии $L_n^{(1)}$ может быть хотя бы одна соседняя точка. Одна – для концевых точек линии, три – для точек на ответвлении, четыре – на пересечении линий и две – в середине линии в остальных случаях.

Сформулированная задача выделения и утоньшения линий на скоростемерных лентах (1) – (8) содержит в себе 80…90% решения задачи разработки и реализации соответствующих алгоритмов и программ. Однако такой подход применим исключительно к идеальным линиям на идеальном носителе. В реальности целый ряд дефектов изображения (табл. 1) значительно усложняет как саму задачу, так и методы ее решения.

При этом решения задач предварительной обработки (повышения качества изображения), выделения и идентификации линий оказываются взаимосвязанными, так как качество предварительной обработки и выделения линий можно считать удовлетворительным только в случае, когда оно позволяет правильно идентифицировать линии. В то же время идентифицировать линии возможно только при качественной предварительной обработке и выделении линий.

Таблица 1 – Дефекты изображения

№	Дефект	Причины	Последствия	Влияние
Дефекты на бумажном носителе				
1	Следы изгибов	После сильного изгиба бумаги на ней остается след, который при сканировании виден из-за отбрасывания тени	Изгиб может восприниматься как вертикальная линия	Значительное
2	Пятна	Загрязнения, дефекты печати, механические повреждения бумаги	В зависимости от расположения, пятно может быть воспринято как линия, либо может помешать при оценке соотношения цветов на участке изображения	От отсутствия до сильного
3	Неоднородность бумаги	Низкое качество бумаги, неправильное хранение	Ухудшение оценки цвета фона, в сочетании с плохой прорисовкой линий может быть воспринято в качестве линии	Среднее
4	Тени	Неровная бумага и расположение лампы подсветки	Снижает контраст между фоном и линией	Среднее
5	Разрывы линии	Неисправность скоростемера, низкое качество бумаги, вмешательство в работу скоростемера	Отсутствие информации о движении в месте разрыва	Значительное
6	Посторонние пометки	Дефект, вносимый человеком	Пометка может быть воспринята как линия	Среднее
Дефекты сканирования				
7	Светлые контуры объектов разметки		Меняется цвет линии, попавшей на контур	Среднее
8	Цветной шум	Шумы цифрового датчика сканера	Ухудшает распознавание разметки	Слабое
9	Переэкспонированные и недоэкспонированные участки	Ограниченный диапазон яркости	Полная потеря информации о линиях на таких участках	Слабое (дефект встречается редко)
Дефекты обработки				
10	Разрывы линий	Неправильный выбор порога фон/линия, пересечение с другими линиями	Потеря информации о линии	Значительное
11	Искажения на краях линий	Неправильный выбор порога фон/линия, округления	Ухудшает определение середины линии	Среднее

Чтобы разорвать этот замкнутый круг предлагается использовать образцы или шаблоны выделенных линий и адаптировать к ним фильтры предварительной обработки и выделения линий в их совокупности.

Целью данной работы является исследование возможностей адаптивной фильтрации изображений скоростемерных лент с применением образцов (или шаблонов) выделенных участков линий, что предусматривает разработку соответствующих методов и программных средств.

Показатель качества фильтрации

Для выполнения адаптации алгоритмов необходимо наличие измерительного инструментария в виде части адаптирующего алгоритма, которая предназначена для оценки качества фильтра. Так как качество фильтра в данном случае связано с его функциональностью и выражается такой характеристикой качества, как функциональная эффективность, показатели качества должны основываться на результатах фильтрации.

Один из подходов оценки качества фильтрации изображения основывается на сравнении с образцом или шаблоном. В нашем случае шаблон может быть получен средствами разработанного авторами функционально-логического подхода к выделению и идентификации линий скоростемерной ленты. Этот подход во многих случаях работает достаточно эффективно, однако требует хорошего качества изображения на всем протяжении ленты.

Определим показатель эффективности фильтра как степень близости результатов фильтрации к шаблону следующим образом.

Пусть в некоторой ограниченной области изображения имеется линия шаблона L^S ($\underline{x}_{ij}^S \in L^S$) и линия, полученная в результате фильтрации L^Φ ($\underline{x}_{ij}^\Phi \in L^\Phi$). Будем также говорить о двух обезличенных линиях L^a и L^b.

С учетом расстояния между точками (2) расстояние от точки $\underline{x}_{ij}^* \in L^a$ до линии L^b:

$$\rho(\overline{x}_{ij}^*, L^b) = \min_{\underline{x}_{kl}^* \in L^b}(\rho(\overline{x}_{ij}^*, \underline{x}_{kl}^*)\,. \tag{9}$$

Обозначим $L_m^{a,b}$ – множество точек линии L^a, удаленных от линии L^b на расстояние $\rho^2 = m$:

$$\underline{x}_{ij}^* \in L_m^{a,b} : \underline{x}_{ij}^* \in L^a \;\&\; \rho^2(\underline{x}_{ij}^*, L^b) = m\,, \tag{10}$$

а также $K_1 = \max\limits_{\nu(L_i^{F,S}) \neq 0}(i)$.

Тогда показатель качества фильтра можно определить как

$$F_1 = \sum_{k=1}^{K_1} \nu(L_k^{\Phi,S}) \cdot k^2\,. \tag{11}$$

Он учитывает степень близости точек линий, полученных в результате фильтрации изображения, к точкам линии шаблона. В лучшем случае, когда все точки фильтра совпадут с точками шаблона, F_1 будет минимальна (равна нулю).

Оценка F_1 будет эффективной, если между точками линии после фильтрации и точками шаблона есть взаимнооднозначное соответствие. Если после фильтрации в линии будет больше точек, чем в шаблоне, то «лишние» ухудшают показатель качества на величину, которая учитывает то, насколько далеко эти точки расположены от линии шаблона. Эта ситуация хорошо отражается в (11).

Ситуация, при которой точек в линии после фильтрации меньше, чем в шаблоне, т.е. линия после фильтрации имеет разрывы, никак не учтена в (11). Для исправления этого недостатка уточним F_1.

Множество точек шаблона, у которых нет соответствующих точек в линии после фильтрации изображения

$$\underline{x}_{ij}^{S} \in \hat{L}^{S} : \underline{x}_{ij}^{S} \in L^{S} \ \& \ (\overline{\exists} \underline{x}_{kl}^{S} \in L^{S} : \underline{x}_{kl}^{S} \neq \underline{x}_{ij}^{S} \ \& \ \rho(\underline{x}_{kl}^{S}, L^{F}) < \rho(\underline{x}_{ij}^{S}, L^{F})), \qquad (12)$$

можно разбить на K_2 подмножеств таким образом (подмножества соответствуют отдельным разрывам в линии после фильтрации):

$$\hat{L}^{S} = \bigcup_{k=1}^{K_2} \hat{L}_k^{S} : \hat{L}_q^{S} \bigcap \hat{L}_p^{S} = \varnothing \text{ при } q \neq p \text{ и } \underline{x}_{ij}^{S} \in L_q^{S}, \text{ если}$$
$$\exists \underline{x}_{kl}^{S} \in L_q^{S} : \underline{x}_{kl}^{S} \neq \underline{x}_{ij}^{S} \ \& \ \rho(\underline{x}_{kl}^{S}, \underline{x}_{kl}^{S}) \leq \sqrt{2}. \qquad (13)$$

Показатель F_2 учитывает потери качества результатов фильтров, связанные с разрывами линий, полученных в результате фильтрации изображения:

$$F_2 = \sum_{k=1}^{K_2} \nu^2(\hat{L}_k^{S}). \qquad (14)$$

Окончательно показатель качества фильтрации изображения в сравнении с шаблоном определим так:

$$F = F_1 + F_2 = \sum_{k=1}^{K_1} \nu(L_k^{\Phi,S}) \cdot k^2 + \sum_{k=1}^{K_2} \nu^2(\hat{L}_k^{S}). \qquad (15)$$

Структура адаптивного алгоритма

Структуру адаптивного алгоритма, согласно нотации алгоритмических структур [9], можно представить следующим образом:

$$Str(A|_{C_0}^{C_{IK}}) = A_1|_{C_0,U_1}^{C_1} \cdot A_2|_{C_1,U_2}^{C_2} \ldots A_{IK}|_{C_{IK-1},U_{IK}}^{C_{IK}} = \prod_{k=IN}^{IK} A_i|_{C_{k-1},U_k}^{C_k}, \qquad (16)$$

где $A_i|_{C_{k-1},U_k}^{C_k}$ – программные фильтры с исходным изображением C_{k-1} (в формате Grayscale), результирующим C_k и управляющими параметрами U_k. C_0 и C_{IK} – начальное и результирующее изображения адаптивного фильтра соответственно.

Количество фильтров находится в заданном интервале $IN \ldots IK$ и фактически является переменным благодаря наличию пустого алгоритма (фильтра).

Образующими алгоритмами $A_k|_{C_{k-1},U_k}^{C_k}$ для формирования структуры адаптивного алгоритма являются апертурные и не апертурные фильтры, приведенные далее.

Алгоритмы фильтрации последовательно просматривают точки исходного изображения $c_{ij}^{k} \in C_{k-1}$ и вычисляются их новые значения $c_{ij}^{k+1} \in C_k$. Относительное положение пикселов, которые используются в качестве аргументов при вычислении нового значения текущего пиксела, задается апертурой.

Большинство используемых образующих фильтров имеют прямоугольную апертуру размерности $N \times M$. При этом вычисляемым является пиксел в центре апертуры, а базовым [5] – пиксел в левом верхнем углу. Не апертурные фильтры можно считать частным случаем апертурных с размерностью апертуры 1×1.

При линейной апертуре формула преобразования имеет вид [6]:

$$c_{ij}^{k+1} = (\sum_{\Omega} w_{ij} c_{ij}^{k}) / d \,, \tag{17}$$

где Ω – апертура, i, j – индексы относительно базового пиксела, w_{ij} – весовые коэффициенты (матрица W), d – нормирующий коэффициент.

Используемые образующие фильтры [4-8], [10] модифицированы с учетом переменной толщины линии. По визуальным эффектам (представлены на рис. 1) их можно разделить на четыре группы.

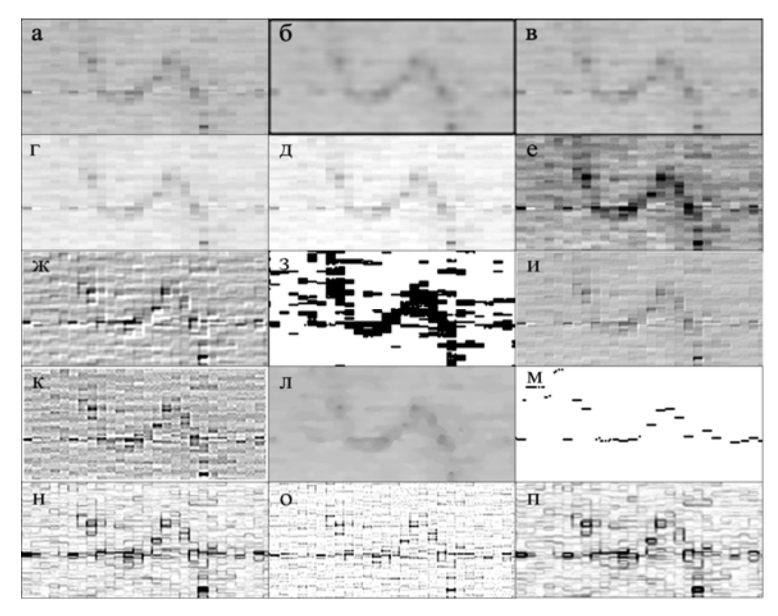


Рисунок 1 – Визуальные эффекты базовых фильтров (а – начальное изображение, б…п – обработанные фильтром)

Первая группа состоит из фильтров предварительной обработки изображения:

– сглаживание устраняет шумы и неровности линий (рис. 1б). Вычисляется согласно (17) с коэффициентами

$$w_{ij} = 1 \,,\ d = (N-1)\cdot(M-1)/4$$

и управляющими параметрами: размер апертуры: $N \times M$ и ограничениями на них N, M – нечетные, $3 \le N \le 15$, $3 \le M \le 15$;

– гауссовское сглаживание (рис. 1в), аналогичное предыдущему. Вычисляется согласно (17) с коэффициентами

$$w_{ij} = \left(\frac{M-1}{2} - \left| \frac{M-1}{2} - i \right| \right) \cdot \left(\frac{N-1}{2} - \left| \frac{N-1}{2} - j \right| \right),\ d = (N+M-2)^2$$

и управляющими параметрами, как в предыдущем фильтре;

– изменение яркости (рис. 1г) не апертурный фильтр со следующим пересчетом значений пикселов

$$c_{ij}^{k+1} = \begin{cases} c_{ij}^{k} \pm l, \text{ при } \underline{c} < c_{ij}^{k} \pm l < \overline{c} \\ \underline{c}, \text{ при } c_{ij}^{k} - l < \underline{c} \\ \overline{c}, \text{ при } c_{ij}^{k} + l > \overline{c} \end{cases}$$

и управляющими параметрами: l – приращение яркости $\underline{c}, \overline{c}$ – минимально максимально возможное значение цвета;

– изменение контрастности (рис. 1д) также не апертурный фильтр,

$$c_{ij}^{k+1} = \begin{cases} c_{ij}^{k} \times l, \text{ при } \underline{c} < c_{ij}^{k} \times l < \overline{c} \\ \underline{c}, \text{ при } c_{ij}^{k} \times l < \underline{c} \\ \overline{c}, \text{ при } c_{ij}^{k} \times l > \overline{c} \end{cases},$$

с аналогичными управляющими параметрами;

– расширение диапазона яркости (рис. 1е)

$$c_{ij}^{k+1} = \underline{c} + \frac{c_{ij}^{k} - c_{min}^{k}}{c_{max}^{k} - c_{min}^{k}} \cdot (\overline{c} - \underline{c}),$$

где c_{min}^{k}, c_{max}^{k} – минимальное максимальное значение цвета в изображении C_k. Управляющие параметры $\underline{c}, \overline{c}$;

– максимальный рельеф (рис. 1ж) повышает контрастность границ линии. Вычисляется согласно (17) с коэффициентами

$$w_{ij} = \begin{cases} 1, \text{ при } i = j = (N+1)/2 \\ (-1)^{p} sign(N-i-j+1)((j-(N+1)/2)^2 + (i-(N+1)/2)^2) \text{ при } 0 \le p \le 1 \\ (-1)^{p} sign(i-j)((i-(N+1)/2)^2 + (j-(N+1)/2)^2) \text{ при } 2 \le p \le 3 \\ 0, \text{ во всех остальных случаях} \end{cases}, \ d = 1$$

и управляющими параметрами: размер апертуры: $N \times N$, направление p и ограничениями на них N – нечетное, $3 \le N \le 15$, $0 \le p \le 3$.

Фильтры выделения линий с исходной толщиной:

– пороговый фильтр со стохастической апертурой (рис. 1з). В заданной области $N \times M$ случайным образом отбирается N^* точек, определяется уровень цвета пикселов τ заданной части пикселов (T в %) с минимальным уровнем цвета. Фильтр преобразует изображение в двухцветное:

$$c_{ij}^{k+1} = \begin{cases} \underline{c}, \text{ при } c_{ij}^{k} < \tau \\ \overline{c}, \text{ при } c_{ij}^{k} \ge \tau \end{cases}$$

и управляющими параметрами $0{,}05 \cdot N \cdot M \le N^* \le 0{,}2 \cdot N \cdot M$ и $2\% \le T \le 6\%$;

– повышение резкости (рис. 1и) увеличивает контраст между соседними пикселами. Вычисляется согласно (17) с коэффициентами

$$w_{ij} = \begin{cases} (n+1)/2, \text{ при } i = j = N/2 \\ -1, \text{ при } i = j \text{ и } i < N/2 \text{ и } p = 0 \\ -1, \text{ при } i = j \text{ и } i > N/2 \text{ и } p = 1 \\ -1, \text{ при } i = N-j+1 \text{ и } i < N/2 \text{ и } p = 2 \\ -1, \text{ при } i = N-j+1 \text{ и } i > N/2 \text{ и } p = 3 \\ 0, \text{ во всех остальных случаях} \end{cases}, \ d = 1$$

и управляющими параметрами: размер апертуры: $N \times N$, направление p и ограничениями на них N – нечетное, $3 \le N \le 15$, $0 \le p \le 3$;

– максимальное повышение резкости (рис. 1к). Вычисляется согласно (17) с коэффициентами

$$w_{ij} = \begin{cases} 2(N-1)+1, \text{при} i = N/2, j = N/2 \\ -1, \text{ при } i = N/2 \text{ или } j = N/2 \text{ и } i \neq j \\ 0, \text{во всех остальных случаях} \end{cases}, \ d = 1$$

и управляющими параметрами, как у сглаживания;

– нелинейный медианный горизонтальный фильтр (рис. 1л) (и аналогичный вертикальный)

$$c^{k+1}_{i,j+r+N/2} = \begin{cases} \text{медиана}(c^k_{i,j+r},\ldots,c^k_{i,j+r+N}), \\ \qquad \text{если медиана}(c^k_{i,j+r},\ldots,c^k_{i,j+r+N}) - \text{медиана}(c^k_{i,j},\ldots,c^k_{i,j+N}) > \tau \\ \qquad \text{и медиана}(c^k_{i,j+r},\ldots,c^k_{i,j+r+N}) - \text{медиана}(c^k_{i,j+2r},\ldots,c^k_{i,j+2r+N}) > \tau \\ c^k_{i,j+r+N/2}, \text{в остальных случаях} \end{cases}$$

с управляющими параметрами $3 \le N \le 15$ и $2 \le r \le 15$.

Фильтры выделения и утоньшения линий:

– выделение середины линий (на рис. 1 не приведен, так как к начальному изображению в формате Grayscale не применим). Уменьшает толщину линии до 1 – 2 пикселов. Вычисляется согласно (8) и с управляющими параметрами;

– нелинейный горизонтальный фильтр (рис. 1м) (и аналогичные диагональные и вертикальный)

$$c^{k+1}_{i,j+N} = \begin{cases} \underline{c}, \text{ если } c^k_{i,j+N} = \min(c^k_{i,j},\ldots,c^k_{i,j+N}) = \min(c^k_{i,j+N},\ldots,c^k_{i,j+2N-1}) \\ c^k_{i,j+N}, \text{ в остальных случаях} \end{cases}$$

с управляющими параметрами $3 \le N \le 15$ и $2 \le j \le 40$.

Линейные фильтры выделения границ линий представлены фильтрами на основе методов (без управляющих параметров):

– Робертса (рис. 1н) $c^{k+1}_{11} = \overline{c} - \sqrt{(c^k_{11} - c^k_{22})^2 + (c^k_{21} - c^k_{12})^2}$;

– Лапласа с инверсией цвета (рис. 1о) $W = \begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}$;

– Собела (рис. 1п)

$$c^{k+1}_{22} = \overline{c} - \sqrt{((c^k_{13} + 2c^k_{23} + c^k_{33}) - (c^k_{11} + 2c^k_{21} + c^k_{31}))^2 + ((c^k_{11} + 2c^k_{12} + c^k_{13}) - (c^k_{31} + 2c^k_{32} + c^k_{33}))^2}.$$

Адаптирующий алгоритм

Задача адаптирующего алгоритма заключается в том, чтобы найти такую последовательность фильтров (16) и, вместе с тем, такие управляющие параметры фильтров применительно к такой последовательности, чтобы показатель качества (15) был минимальным.

Адаптирующий алгоритм состоит из двух генетических алгоритмов: один подбирает цепочку фильтров (ГАЦФ), второй – подбирает параметры для конкретной цепочки (ГАП).

Для работы алгоритма ГАЦФ составляется хромосома, представляющая из себя последовательность графических фильтров. Каждый фильтр имеет свой набор параметров – последовательность целых чисел, размер которой и назначение элементов

может меняться в зависимости от фильтра. Отличием от классического генетического алгоритма является использование в качестве генов не двоичных чисел, а составных структур, которые могут существенно отличаться друг от друга. Это заставляет внести в работу алгоритма условие, по которому операция кроссовера возможна лишь между генами одного и того же типа. Необходимость этого условия заключается в том, что разные фильтры по-разному интерпретируют значения параметров, а также для разных фильтров накладываются разные ограничения на допустимые значения параметров. Оценка хромосомы этого алгоритма – F, вычисленная согласно (15) по параметрам заданной в хромосоме последовательности фильтров, подобранным ГАП.

Алгоритм ГАП работает с готовой последовательностью фильтров, и менять их расположение или состав не может. Его задача – подобрать такие значения параметров этих фильтров, при которых применение этой последовательности к изображению даст минимальное значение функции (15). Полученная оценка принимается за оценку цепочки фильтров. Хромосомой является последовательность всех параметров всех фильтров в заданной ГАФЦ последовательности фильтров.

На каждом уровне выполняются действия, характерные классическим генетическим алгоритмам [11], – формируется популяция хромосом, производится их оценка, модификация, переход к следующему поколению.

Преимущество двухуровневого алгоритма в том, что он не только позволяет подобрать параметры к фильтрам, но и последовательность их применения. Это дает возможность задействовать множество простых фильтров, которые не дают нужного результата по отдельности, но в комбинации с другими могут значительно приблизить исходное изображение к эталонному.

На рис. 2 приведен один из шаблонов, к которому адаптирован фильтр (рис. 2а), область фильтрации (область шаблона и соседняя контрольная область на рис. 2б) и результаты фильтрации (рис. 2в). Адаптивный алгоритм показал удовлетворительные результаты, несмотря на то, что базовые фильтры по отдельности не эффективны (рис. 1).

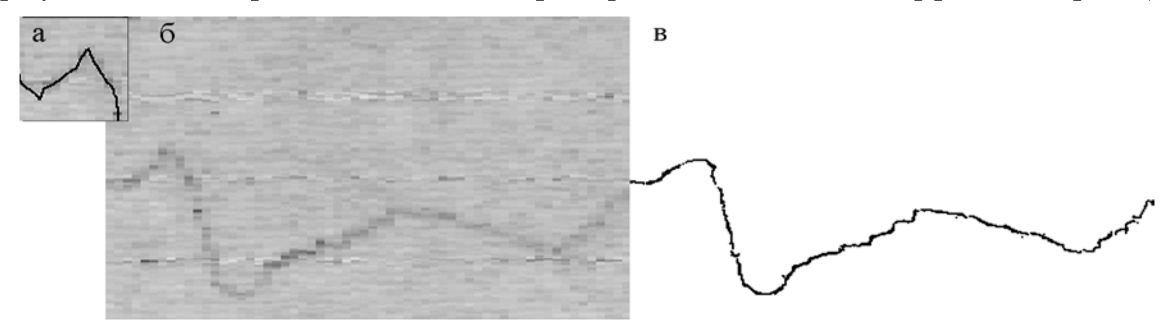


Рисунок 2 – Пример применения адаптивного фильтра

Выводы

Применение адаптивной фильтрации скоростемерных лент позволило значительно улучшить качество выделения линий толщиной в один пиксел, что существенно отразилось на решении задач выявления нарушений режима и требований правил управления локомотивом.

Автоматизация процесса обработки скоростемерных лент в целом позволяет формировать информационную базу маршрутов, поездок, нарушений и неисправностей. На основании анализа этой информации можно решать задачи, связанные с экономией энергоресурсов, оптимизацией работ по ремонту пути, выявлением узких мест в организации движения поездов, обучением машинистов и др., что и определяет перспективу дальнейших исследований.

Литература

1. Венцевич Л.Е. Локомотивные скоростемеры и расшифровка скоростемерных лент : учеб. пособие / Венцевич Л.Е. – М.: УМК МПС России, 2002. – 272 с.
2. Програмно-апаратний комплекс розшифровки швидкостемірних стрічок / В.І. Шинкаренко, Є.Г. Васецький, Т.М. Мажара, О.М. Швець // Современные информационные технологии на транспорте, в промышленности и образовании : Междунар. науч.-практ. конф., 15-16 мая 2008 г. : тезисы докл. – Д.: ДНУЖТ, 2008. – С. 80-81.
3. Мухин О.С. Разработка методов автоматизированной обработки скоростемерных лент поездок локомотивных бригад : автореферат дис. ... канд. техн. наук : 05.22.07 : «Подвижной состав железных дорог, тяга поездов и электрификация» / О.С. Мухин. – Омск, 2003. – 21 с.
4. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
5. Яншин В.В. Обработка изображений на языке СИ для IBM PC: Алгоритмы и программы / В.В. Яншин, Г.А. Калинин. – М. : Мир, 1994. – 241 с.
6. Прэтт У. Цифровая обработка изображений / У. Прэтт. – М. : Мир, 1982. – Кн. 1 – 312 с.; Кн. 2 – 480 с.
7. Форсайт Д.А. Компьютерное зрение. Современный поход / Д.А. Форсайт, Ж. Понс. – М. : Издательский дом «Вильямс», 2004. – 928 с.
8. Методы компьютерной обработки изображений / [под ред.. В.А. Сойфера]. – 2-е изд. – М. : Физматлит, 2003. – 784 с.
9. Шинкаренко В.И. Структурные модели алгоритмов в задачах прикладного программирования Часть I : Формальные алгоритмические структуры / В.И. Шинкаренко, В.М. Ильман, В.В. Скалозуб // Кибернетика и системный анализ. – 2009. – № 3. – С. 3-14.
10. Бейтс Р. Восстановление и реконструкция изображений / Р. Бейтс, М. Мак-Доннелл. – М. : Мир, 1989. – 336 с.
11. Гладков Л.А. Генетические алгоритмы : учебное пособие / Л.А Гладков, В.В. Курейчик, В.М. Курейчик. – 2-е изд. – М. : Физматлит, 2006 – 320 с.

Literatura

1. Vencevich L.E. Locomotive speedometers and speedometer tape interpretation: Tutorial. – 2002.
2. Shinkarenko V.I., Vasetskiy E.G., Mazhara T.M., Shvets O.M. Modern information technologies on transport, industry and education. – 2008.
3. Muhin O.S. Development of methods for automated processing of speedometer tapes taken on tours of locomotive drivers: synopsis. – 2003.
4. Gonsalez R., Woods R. Digital image processing. – 2005.
5. Yanshin V.V., Kalinin G.A Image processing on C language for IBM PC.Algorithms and programs. – 1994.
6. Prett W. Digital image processing. – 1982.
7. Forsyth D.A., Ponce J. Computer vision: A modern approach. –2004.
8. Soyfer V.A. Methods of computer image processing. – 2003.
9. Shinkarenko V.I., Il'man V.M., Skalozub V.V. Cybernetics and system analysis. – 2009.
10. Bates R., McDonnell M. Image restoration and reconstruction. – 1989.
11. Gladkov L.A., Kureychik V.V., Kureychik V.M. Genetic algorithms: Tutorial. – 2006.

В.І. Шинкаренко, Є.Г. Васецкий, І.Д. Корякін

Адаптивна фільтрація зображень швидкостемірних стрічок локомотивів

У статті представлені розроблені методи та засоби структурної адаптації алгоритмів структурної адаптації алгоритмів програмної фільтрації зображень. Вирішується задача виокремлення та потоншення ліній швидкостемірних стрічок локомотивів. З застосуванням генетичних алгоритмів підбирається послідовність базових фільтрів та їх параметрів, адаптована до заданих шаблонів ліній за запропонованим критерієм.

V.I. Shynkarenko, E.G. Vasetsky, I.D. Koryakin

Classifiers Construction Based on Separate Hyper Surfaces

This paper represents developed methods and tools for structure adaptation of software image filtering algorithms. The problem of allocation and thinning lines on locomotive speedometer tape is being solved. With an application of genetic algorithms sequence of basic filters and their parameters is being picked, which is adapted to given lines templates and corresponds to the proposed criterion.

Статья поступила в редакцию 22.06.2011.